

SOA-Governance

Serviceorientierte Architekturen (SOA) tragen zur Steigerung der Komplexität betrieblicher IT-Strukturen bei. Ohne geeignete Kontroll- und Steuerungsstrukturen kann die resultierende Komplexität zu Strukturen führen, deren Wartung und Pflege ähnlich aufwendig wird wie die der bestehenden evolutionären Applikationslandschaften, die sie ablösen sollen. Um dies zu verhindern, ist der Aufbau einer ganzheitlichen SOA-Governance unumgänglich.

Inhaltsübersicht

- 1 Einführung
- 2 Serviceorientierung, Wirkung und Grenzen
- 3 Notwendigkeit von SOA-Management und -Governance
- 4 Beispielhafte Entwicklung eines SOA-Governance-Modells
 - 4.1 SOA-Managementaktivitäten
 - 4.2 SOA-Organisationsstruktur
- 5 Fazit
- 6 Literatur

1 Einführung

Zur Gestaltung des komplexen Systems Unternehmung werden üblicherweise unterschiedliche Abstraktionsebenen gebildet, wobei in der Regel die Ebenen Geschäftsarchitektur, Prozessarchitektur, Integrationsarchitektur und Softwarearchitektur unterschieden werden können [Winter & Fischer 2006].

Entsprechend dem Vorrang der Fachbereiche gegenüber der IT kann eine hierarchische Gestaltungsfolge angenommen werden: Aus der strategischen Positionierung des Unternehmens in den Geschäftsmodellen werden geeignete Aufbau- und Ablauforganisationsformen abgeleitet. An ihrer Unterstützung orientiert sich die Gestaltung des Informationssystems.

An dieser Stelle macht sich jedoch ein Zielkonflikt bemerkbar. Die fachliche Sicht auf das Unternehmen muss auf die Agilität des Unternehmens abstellen und passt dementsprechend Aufbau- und Ablauforganisation in schneller Folge daran an. Demgegenüber stehen die IT-Systeme, die in der evolutionär entstandenen Form deutlich langsamer entwickelt und verändert wurden. Aus der nur langsamen Änderbarkeit früher entwickelter Systeme entstand in der Konsequenz eben diejenige Applikationslandschaft, in der Kernsysteme fortlaufend um weitere Systeme ergänzt wurden, da die Erweiterung der Kernsysteme typischerweise mehr Zeit benötigt hätte. Um die sich (potenziell) schneller ändernden Organisationsstrukturen und die sich langsamer ändernden Softwarestrukturen miteinander in Einklang zu bringen, wird zwischen den entsprechenden Architekturebenen explizit die Integrationsebene betrachtet, die gleichsam eine logische Übersetzung der Artefakte der Prozessarchitektur in IT-nahe Artefakte darstellt und zu einer Einkopplung der unterschiedlich getakteten Ebenen beiträgt [Braun & Winter 2005; Schelp & Winter 2007]. Prägnante Praxisbeispiele für die Einführung einer derartigen Integrationsebene sind Unternehmen wie die Deutsche Post [Herr et al. 2004], die Credit Suisse [Hagen 2003] oder Hersteller wie SAP mit ihrer Enterprise Service Architecture (ESA) (vgl. z. B. [Duan et al. 2005], [Heutschi et al. 2006]).

2 Serviceorientierung, Wirkung und Grenzen

Services können auf unterschiedlichen Ebenen identifiziert werden [Winter & Fischer 2006]. Bezogen auf die Diskussion serviceorientierter Architekturen sind zwei Architekturebenen von

besonderer Bedeutung: Enterprise Services auf der Integrationsebene kapseln IT-unterstützbare fachliche Funktionalität in Form von Komponenten, die lose gekoppelt sind. Sie können wiederum aus Enterprise Services feinerer Granularität bestehen. Die elementaren Services finden dann auf der Softwareebene ihr Gegenstück in Form der Software-Services, die als Softwarekomponenten z. B. als Web-Services implementiert sein können.

Unabhängig von der Betrachtungsebene stellt das Designprinzip der losen Kopplung den wichtigsten Unterschied zur herkömmlichen, nicht serviceorientierten Applikationsentwicklung dar: Durch die lose Kopplung der Komponenten soll es ermöglicht werden, Änderungen in den zu unterstützenden Geschäftsprozessen eher durch Rekombination der verwendeten Services als durch Änderungen der implementierten Funktionalität abbilden zu können.

Ähnlich wie bei der Diskussion der Enterprise Application Integration (EAI) erfährt die Diskussion der Serviceorientierung häufig eine Reduktion auf eher technische Aspekte wie die Schnittstellenkomplexität. Fallstudien zeigen, dass durch lose Kopplung tatsächlich eine Reduktion der Schnittstellenkomplexität in umfangreichen Applikationslandschaften erreicht werden kann [Hagen 2003; Puschmann et al. 2002]. Jedoch verringert dies nicht unbedingt die Komplexität der Abhängigkeitsbeziehungen. Die Komplexität kann sogar steigen: Wenn neuentwickelte Services (und die sie implementierenden Softwarekomponenten) kleiner als klassische Applikationen geschnitten sind, um eine flexible Kopplung zu größeren Services bzw. Applikationen zu ermöglichen, müssen entsprechend mehr Services gebaut werden, wodurch die Anzahl der einzelnen Softwarekomponenten und ihrer Schnittstellen steigt. Überdies müssen diese Schnittstellen und die damit realisierten Verbindungen explizit betrachtet werden, wohingegen viele dieser Verknüpfungen zuvor in den größeren Applikatio-

nen gebündelt wurden und nur bei deren lokaler Änderung zu berücksichtigen waren.

Obwohl serviceorientierte Architekturprinzipien hinsichtlich verkürzter Umsetzungszeit und höherer Flexibilität des Gesamtsystems von Vorteil sind, müssen doch zwei negative Konsequenzen berücksichtigt werden. Zum einen verringert das Vorhalten produktiver Varianten die Anzahl der Wiederverwendungen, die die einzelne Variante eines Service erfahren kann. Tatsächlich ist der erzielte Wiederverwendungsgrad bei umfangreicheren Implementierungen serviceorientierter Architekturen nicht beeindruckend: In Einzelfällen wird bei Applikationslandschaften mit mehreren Hundert produktiven Services von Wiederverwendungsgraden kleiner als zwei berichtet. Gleichzeitig wird dabei angegeben, dass der Aufwand, einen Service tatsächlich wiederverwendbar zu gestalten, um mehr als das Dreifache höher eingestuft wird, als ihn nicht wiederverwendbar zu gestalten.

Zum anderen steigt die Komplexität des Gesamtsystems, und es besteht die Gefahr, dass die Abhängigkeiten zwischen den Services nicht mehr überblickt werden können bzw. kaum noch auflösbare Abhängigkeitsnetze entstehen: beispielsweise wenn neue Services weiter mit Bezug auf ältere existierende Servicevarianten gebaut werden.

Eine Grenze erfährt die schnellere Implementierbarkeit von Änderungen zudem, wenn die maximal zulässige Anzahl produktiver Varianten erreicht ist. Ist dieser Spielraum ausgeschöpft, muss eine Variante – typischerweise die älteste Servicevariante – außer Dienst gestellt werden und die durch die Varianten entstandenen Signaturänderungen in den beziehenden Services/Applikationen nachgezogen werden, was nicht selten auch dort umfangreichere Änderungen zur Folge haben kann.

Um die skizzierten Abhängigkeitsbeziehungen vollständig nachvollziehen zu können, ist die Verwendung entsprechender Repositories erforderlich. Um die Beziehungsgeflechte voll-

umfänglich zu erfassen, ist zudem ein umfangreiches Metamodell erforderlich, das die einzelnen Artefakttypen auf den verschiedenen Ebenen berücksichtigt. Der in Abbildung 1 dargestellte Ausschnitt eines solchen Metamodells definiert Enterprise Services als fachliche Services auf der Integrationsebene. Sie repräsentieren Aktivitäten oder Aktivitätsbündel der Geschäftsprozesse aus der Organisationsebene und können unterschieden werden in manuelle und automatische Aktivitäten. Letztere wieder lassen sich über die fachlichen Services der Integrationsebene auf Softwarekomponenten abbilden, die auf der Software- und Datenbankebene angesiedelt sind. Die in ihnen enthaltenen Funktionen reflektieren die einzelnen Aktivitäten der Geschäftsprozesse.

Um die angesprochenen Beziehungsgeflechte hinsichtlich fachlicher Änderungsanforderungen richtig beurteilen zu können, ist es von zentraler Bedeutung, nicht nur die technische Perspektive (Softwarekomponenten), sondern auch die Geschäftsprozesse im Detail zu erfassen und entsprechende Repositories mit fachlichen wie softwaretechnischen Informationen zu versehen. Bei einer serviceorientierten Architektur ist eine ganzheitliche Architektursicht notwendig, die nur im Rahmen einer Enterprise Architecture erzielt werden kann.

Allein durch die Existenz eines solchen Repositories können die beschriebenen Abhängigkeitsgeflechte jedoch nicht eingegrenzt und kontrolliert werden. Es sind spezifische organisatorische Maßnahmen erforderlich, die in dem bestehenden Architekturmanagement be-

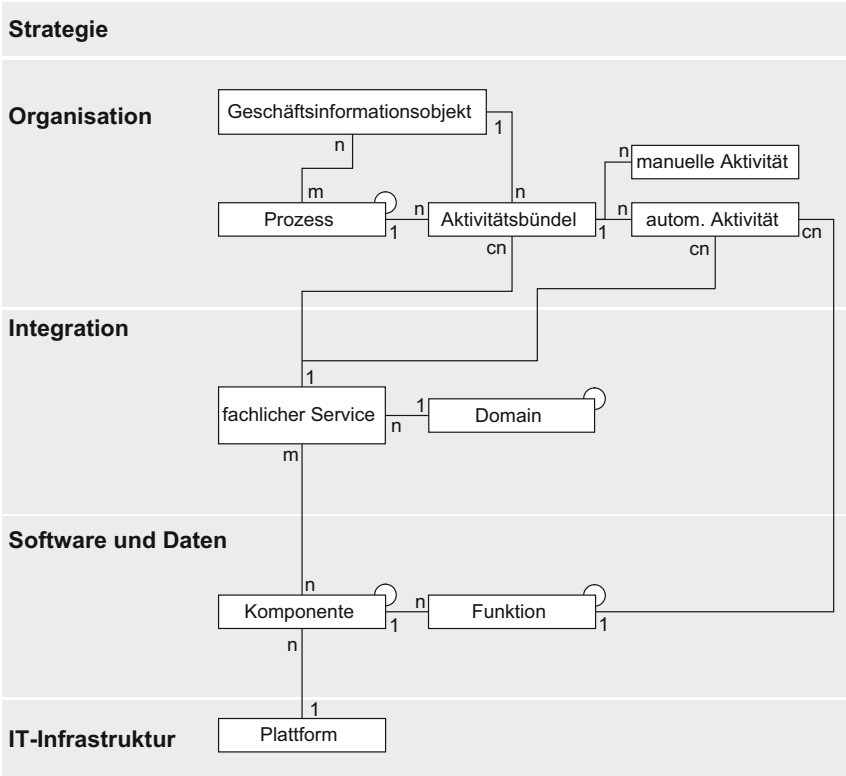


Abb. 1: Beispielhaftes SOA-Metamodell

rücksichtigt und um SOA-spezifische Aspekte ergänzt werden müssen – insbesondere ergeben sich Anforderungen an die SOA-Governance.

3 Notwendigkeit von SOA-Management und -Governance

Das IT-Architekturmanagement beinhaltet die Grundfunktionen der langfristigen Architekturführung, Architekturweiterentwicklung und Architekturvertretung im Unternehmen [Hafner & Winter 2004]. Die zuvor genannten Probleme des Wachstums der Komplexität einer serviceorientierten Architektur erfordern das Festlegen von Designvorschriften, Implementierungsstandards und Controlling-Mechanismen, um das Wachstum langfristig kontrollieren und so Ineffizienzen vermindern zu können. Über alle Phasen der Umsetzung fachlicher Anforderungen in Services hinweg ist ein übergeordnetes Management notwendig, das das Zusammenspiel der Services koordiniert und steuert: Analog zur IT-Governance [Schelp et al. 2006] ist die Notwendigkeit einer SOA-Governance unbestritten [Malinverno 2006a]. Dabei gilt es, die Grundprinzipien der IT-Governance [Weill & Ross 2004] im Kontext der Enterprise Governance auf die serviceorientierten Architekturen anzuwenden (vgl. Abb. 2).

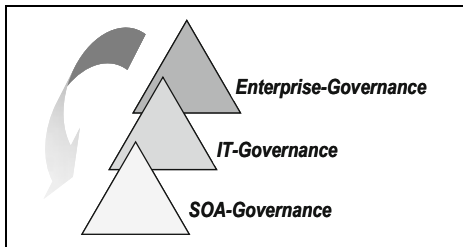


Abb. 2: Einbettung der SOA-Governance im Unternehmenskontext

Somit definiert sich SOA-Governance als evolutionäre Weiterentwicklung bzw. Spezialisierung der IT-Governance [Malinverno 2006b; Shah et al. 2006], indem adäquate Mechanismen zur Entscheidungsfindung festgelegt werden [Ma-

linverno 2006a], die dazu beitragen, dass auch die SOA das Erreichen der festgelegten Unternehmensziele ganzheitlich unterstützt [Weill & Ross 2004]. Dies wird hauptsächlich über Vorgaben betreffend Organisationsstrukturen, Standards, Rollen, Verantwortlichkeiten und Messmetriken geregelt, die die Transparenz und Konformität serviceorientierter Architekturen sicherstellen. Der zukünftige Nachweis des Einhaltens beispielsweise regulatorischer und gesetzlicher Anforderungen auch im Rahmen der implementierten Unternehmensarchitekturen wird so ebenfalls erleichtert.

4 Beispielhafte Entwicklung eines SOA-Governance-Modells

Insgesamt umfasst ein SOA-Governance-Modell eine Kombination aus Managementaktivitäten und Organisationsstrukturen, die auf Regeln und Vorgaben, den sogenannten Governance Principles [Shah et al. 2006], beruhen und damit die strategische Ausrichtung des Gesamtvorhabens garantieren (vgl. Abb. 3). Dabei umfassen die Managementaktivitäten die »harten« Faktoren und beantworten die Frage: »Was ist jeweils zu tun?« Im Gegenzug decken die Organisationsstrukturen die »weichen« Faktoren ab und behandeln die Frage: »Wer macht was?« Beide Seiten zusammen, die Managementaktivitäten und die Organisationsstrukturen, beantworten die Frage nach dem »Wie wird es gemacht?«.

4.1 SOA-Managementaktivitäten

Das SOA-Management umfasst alle Aktivitäten, die eine erfolgreiche SOA-Umsetzung erfordern, d.h. angefangen bei der Planung über die Implementierung, der Wartung bis hin zum Controlling. Im Folgenden werden die Aktivitäten in den drei Gruppen *Implement*, *Manage* und *Control* zusammengefasst.

- *Implement* SOA: Als Erstes gilt es, im Unternehmen die geeigneten Voraussetzungen für eine SOA-Einführung zu schaffen (*SOA Enab-*

- lement). Danach wird festgelegt und priorisiert, welche SOA-Projekte durchgeführt werden (*SOA Portfolio Management*). Anschließend werden die Designrichtlinien festgelegt (*SOA Service Design*), bevor der Service gemäß den bestehenden Entwicklungsparadigmen umgesetzt wird (*SOA Service Build*).
- **Manage SOA:** Um die Services bzw. ihre Entwicklung und Verwendung steuern und kontrollieren zu können, wird zentral ein Repository geführt (*SOA Repository*), das z. B. auch zum Wiederauffinden eines bestehenden Service genutzt werden kann. Für die Wartung und Pflege der Services (*SOA Maintenance*) ist es hilfreich, auch die Versionsinformationen mitzupflegen – zur Kontrolle der Komplexität der Abhängigkeiten ist dies eine essenzielle Voraussetzung. Schlussendlich können Services über ihre Verbraucher (Consuming Services) transparent den betreffenden fachlichen Nutzern verrechnet werden (*SOA Funding*).
 - **Control SOA:** Zum Controlling einer SOA (*SOA Controlling*) gehört unter anderem, dass die

vereinbarten Leistungs- und Funktionalitätsvorgaben (*SOA Service Level Management*) z. B. anhand von SOA-Kennzahlen jederzeit eingehalten und erfüllt werden. Diese gilt es, im Sinne der Eigner auf Fachseite zu optimieren und stetig zu verbessern (*SOA Quality Management*). Sollte trotzdem unverhofft ein Vorfall auftreten, ist es vorteilhaft, ein *SOA Risk & Security Management* implementiert zu haben. Bezogen auf die Weiterentwicklung der gesamten Architektur ist hier innerhalb der IT zu prüfen, inwieweit den IT-Zielen wie z. B. der Wiederverwendung Rechnung getragen wird.

Die oben genannten Probleme der Abhängigkeitsbeziehungen und des Variantenmanagements gehen über die Fragestellungen einzelner Implementierungsprojekte hinaus, bzw. die übergeordneten Ziele zur Kontrolle der Komplexität der gesamten serviceorientierten Architektur stehen unter Umständen in Konflikt zu den Implementierungszielen der einzelnen Projekte. Beispielsweise wenn die maximale

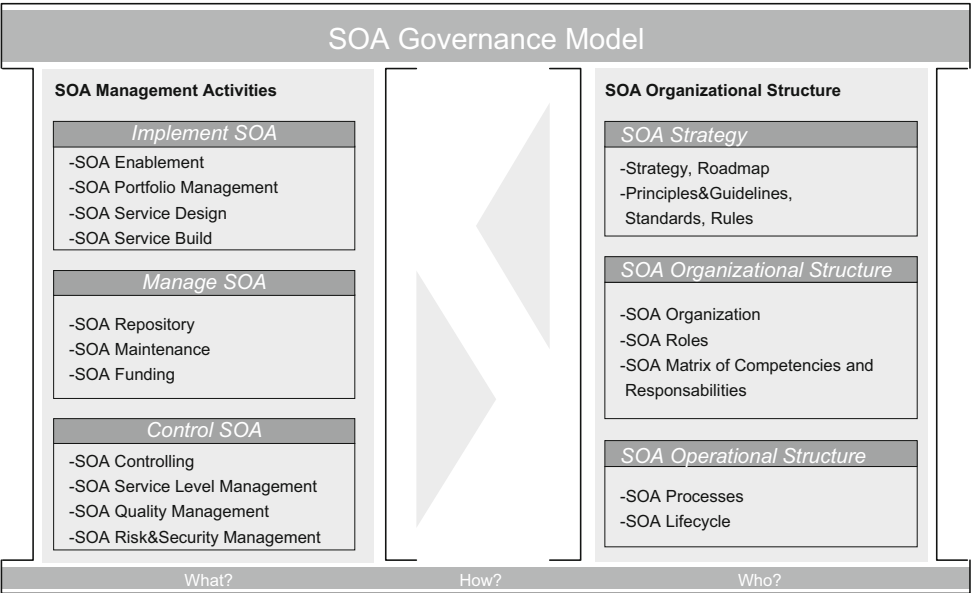


Abb. 3: SOA-Governance-Modell

Anzahl zulässiger Varianten erreicht ist und erst in den abhängigen Services Änderungen vorgenommen werden müssen. Um diesen Fällen zu begegnen und die sich bei Einhalten der Regeln ergebende Verringerung der Umsetzungsgeschwindigkeit zu vermeiden, müssen die SOA-Managementaktivitäten auch ein von den Implementierungsprojekten unabhängiges Überarbeiten der serviceorientierten Architektur beinhalten – sie müssen eigene Veränderungsprojekte anstoßen können.

4.2 SOA-Organisationsstruktur

Grundlage einer jeden serviceorientierten Architektur bildet die *SOA-Strategie*, die beispielsweise festhält, in welchem Ausmaß eine serviceorientierte Architekturgestaltung Einfluss auf die gesamte Unternehmensarchitektur haben soll und welche Geschäftsprozesse tatsächlich im Rahmen einer SOA letztlich auch softwaretechnisch umgesetzt werden sollen. Eine entsprechende Umsetzungsplanung führt dann zu einer *SOA Roadmap*. In dieser wird stets der Kontext zum Gesamtunternehmen hergestellt, um so den Nutzen der Implementierung transparent nachzuweisen. Ebenfalls in der SOA-Strategie sollten alle Prinzipien, Leitlinien, Standards und Regeln verbindlich festgehalten werden, um die Nachhaltigkeit und Kontinuität der Architektur zu gewährleisten. Ihre Einhaltung zu überwachen ist dann Gegenstand der SOA-Managementaktivitäten.

Die Einführung einer SOA zieht auch organisatorische Veränderungen nach sich. So muss innerhalb einer SOA-Governance definiert werden, in welcher organisatorischen Form die SOA-Governance betrieben wird und wie sie sich nahtlos in die bestehende Organisation einbetten kann (*SOA-Aufbauorganisation*) [Carlson & Marks 2006]. Dies kann z. B. zum Aufbau eines SOA Competence Center führen, das zentral und unternehmensweit alle die SOA betreffenden Aktivitäten ausführt. Daneben existieren sogenannte SOA Boards, die sich interdisziplinär mit der langfristigen Planung der

serviceorientierten Architektur auseinandersetzen. Diese Mechanismen können dazu beitragen, die serviceorientierte Architektur langfristig im Unternehmen zu verankern und das Verhalten der involvierten Mitarbeiter zu beeinflussen. Die Besetzung der Boards kann nicht alleine auf die IT-Seite beschränkt bleiben: Analog zum aufgezeigten Metamodell müssen Vertreter der Fachseite einbezogen werden, die die erwarteten Änderungen der Geschäftsprozesse beurteilen können, um so schon frühzeitig Hinweise auf die kommenden Änderungsanforderungen an die definierten Enterprise Services zu erhalten und umgekehrt die Auswirkungen geplanter Änderungen der Services für das Geschäft beurteilen zu können bzw. die Aktivitäten zu priorisieren. Auf Seite der IT sind neben den IT-Verantwortlichen auch die besonderen Interessen von Entwicklung und Betrieb zu berücksichtigen und deren Vertreter einzubinden. Sind Entwicklung und/oder Betrieb ausgelagert, sind entsprechend angepasste Strukturen erforderlich [Schelp et al. 2006].

Daraus ergibt sich die Notwendigkeit, die Rollen einheitlich festzulegen, die im Kontext einer serviceorientierten Architektur in der Unternehmung auszuüben sind. Jede Rolle ist dabei mit einer oder mehreren Funktionen gekoppelt, die spezifische Fähigkeiten verlangen. Diese Rollen lassen sich mit den SOA-Managementaktivitäten koppeln, was z. B. anhand von RACI-Matrizen (Responsible, Accountable, Consulted and Informed) [Malinverno 2006b] oder AKV-Zuordnungen (Aufgabe, Kompetenz, Verantwortung) bewerkstelligt werden kann (vgl. Abb. 4).

So können eindeutig und transparent die Zuständigkeiten und Kompetenzen der SOA-spezifischen Rollen und Aktivitäten festgelegt werden, was die wachsende Komplexität beherrschbar macht.

Neben den oben geschilderten Veränderungen in der Aufbauorganisation hat die Einführung einer SOA auch Auswirkungen auf die Unternehmensprozesse. So müssen neue SOA-

Roles of SOA Participants						
		Business Analyst	Process Manager	Domain Owner	Service Tester	...
SOA Activities	Develop SOA Strategy	<i>R</i>	<i>I</i>	<i>C</i>	<i>I</i>	..
	Maintain SOA Repository	<i>I</i>		<i>A</i>	<i>C</i>	...
	Review SOA Guidelines	<i>I</i>	<i>C</i>	<i>R</i>	<i>R</i>	...

R = Responsibility A = Accountability C = Consult I = Inform						

Abb. 4: Beispielhafte SOA-RACI-Matrix

Prozesse eingeführt, d. h. bestehende Architekturprozesse angepasst werden. Dazu werden die den Prozessen zugrunde liegenden SOA-Aktivitäten häufig anhand eines iterativen *SOA Lifecycle* geordnet.

5 Fazit

Serviceorientierte Architekturen bergen das Potenzial, die betriebliche Applikationslandschaft schneller und flexibler an die sich wandelnden Anforderungen seitens der Fachseite anzupassen. Dies wird ermöglicht durch eine steigende Komplexität. Ohne eine strenge Kontrolle der entstehenden Architektur und rigorose Durchsetzung komplexitätsbegrenzender Regeln kann die sich einstellende Situation schwerer zu handhaben sein als bei den vorgängigen monolithischen Applikationsstrukturen. Wird aber eine solche Kontrolle im Rahmen einer SOA-Governance etabliert und durchgesetzt, kann die betriebliche IT ihrerseits zu allen Aspekten der Agilität des Unternehmens beitragen: kurzfristig im Sinne höherer

Flexibilität und schnellerer Umsetzung, langfristig durch größere Innovationsfähigkeit und – bei fortlaufender Überarbeitung der Architektur – durch Heben der Wiederverwendungspotenziale mittels Begrenzung der kurzfristig zugelassenen Varianten.

Für das erfolgreiche Betreiben einer serviceorientierten Architektur ist eine adäquate SOA-Governance Voraussetzung. Diese erfordert aber ein klares Management-Commitment sowie eine entsprechende Veränderungs- und Investitionsbereitschaft. Kurzfristige Kostenvorteile werden sich jedoch nur schwer erreichen lassen, die über eine auch mit einfacheren Mitteln erreichbare Minderung der Schnittstellenkomplexität im EAI-Sinne hinausgehen.

6 Literatur

[Braun & Winter 2005] Braun, C.; Winter, R.: A Comprehensive Enterprise Architecture Metamodel and Its Implementation Using a Metamodeling Platform. In: Desel, J.; Frank, U. (Hrsg.): Enterprise Modelling and Information Systems Ar-

- chitectures – Proceedings of the Workshop in Klagenfurt, Oktober 24-25, 2005, Bd. P-75, Bonn, 2005, S. 64-79.
- [Carlson & Marks 2006] *Carlson, B.; Marks, E.*: SOA Governance Best Practices – Architectural, Organizational, and SDLC Implications, <http://webservices.sys-con.com/read/175376.htm>, 2006.
- [Duan et al. 2005] *Duan, Z.; Bose, S.; Stirpe, P. A.; Shoniregun, C.; Logvynovskiy, A.*: SOA without Web services: a pragmatic implementation of SOA for financial transactions systems. In: Proceedings of the 2005 IEEE International Conference on Services Computing (SCC2005), July 11-15 2005, Orlando, Florida, Bd. 1, Los Alamitos, CA, USA, 2005, S. 243-250.
- [Hafner & Winter 2004] *Hafner, M.; Winter, R.*: Vorgehensmodell für das Management der Applikationsarchitektur im Unternehmen. 2004.
- [Hagen 2003] *Hagen, C.*: Integrationsarchitektur der Credit Suisse. In: Aier, S.; Schönherr, M. (Hrsg.): Enterprise Application Integration – Flexibilisierung komplexer Unternehmensarchitekturen. Berlin, 2003, S. 61-81.
- [Herr et al. 2004] *Herr, M.; Bath, U.; Koschel, A.*: Implementation of a Service Oriented Architecture at Deutsche Post MAIL. In: Zhang, L.-J.; Jeckle, M. (Hrsg.): Proceedings of the European Conference on Web Services (ECOWS 2004), September 27-30 2004, Erfurt, Germany, Bd. 3250, Berlin et al., 2004, S. 227-238.
- [Heutschi et al. 2006] *Heutschi, R.; Legner, C.; Österle, H.*: Serviceorientierte Architekturen – Vom Konzept zum Einsatz in der Praxis. In: Schelp, J.; Winter, R.; Frank, U.; Rieger, B.; Turowski, K. (Hrsg.): Integration, Informationslogistik und Architektur – Proceedings der DW2006, September 21-22, Friedrichshafen, Germany, Bd. P90, Bonn, 2006, S. 361-382.
- [Malinverno 2006a] *Malinverno, P.*: Sample Governance Mechanisms for a Service-Oriented Architecture. 2006.
- [Malinverno 2006b] *Malinverno, P.*: Service-Oriented Architecture Craves Governance. 2006.
- [Puschmann et al. 2002] *Puschmann, T.; Alt, R.; Sassmannshausen, D.*: Enterprise Application Integration bei Robert Bosch. In: Österle, H.; Fleisch, E.; Alt, R. (Hrsg.): Business Networking in der Praxis – Beispiele und Strategien zur Vernetzung mit Kunden und Lieferanten. Berlin et al., 2002, S. 271-298.
- [Schelp et al. 2006] *Schelp, J.; Schmitz, O.; Schulz, J.; Stutz, M.*: Governance des IT-Sourcing bei einem Finanzdienstleister. In: HMD – Praxis der Wirtschaftsinformatik, 41. Jg, Heft 250, 2006, S. 88-98.
- [Schelp & Winter 2007] *Schelp, J.; Winter, R.*: Towards a Methodology for Service Construction. In: 40th Annual Hawaii International Conference on Systems Sciences (HICSS'07), Hawaii, 2007.
- [Shah et al. 2006] *Shah, R.; Bieberstein, N.; Bose, S.; Fiammante, M.; Jones, K.*: SOA Project Planning Aspects, <http://websphere.sys-con.com/read/168398.htm>, 2006.
- [Weill & Ross 2004] *Weill, P.; Ross, J.*: IT Governance – How Top Performers Manage IT. Boston, 2004.
- [Winter & Fischer 2006] *Winter, R.; Fischer, R.*: Essential Layers, Artifacts, and Dependencies of Enterprise Architecture. In: Society, I. C. (Hrsg.): EDOC Workshop on Trends in Enterprise Architecture Research (TEAR 2006), October 17, 2006, Hong Kong, Los Alamitos, CA, USA, 2006.

Dr. Joachim Schelp
 lic. oec. HSG Matthias Stutz
 Universität St. Gallen
 Institut für Wirtschaftsinformatik IWI-HSG
 Müller-Friedberg-Str. 8
 CH-9000 St. Gallen
 {joachim.schelp, matthias.stutz}@unisg.ch
 www.iwi.unisg.ch